20

Executing

Critical Software Thing: David Gauthier, Audrey Samson, Eric Snodgrass, Winnie Soon, and Magda Tyżlik-Carver

An expectation comes from a technical understanding of execution that it is a straightforward running of tasks within a machine. For instance, in computer science, execution is often associated specifically with the fetch-decode-execute instruction cycle, during which a computer’s central processing unit retrieves instructions from its memory, determines what actions the instructions dictate, and proceeds to attempt to carry out those actions. In this common reading, execution is the performance of such step-by-step instructions.

But of course the instruction cycle does not encompass execution’s impact and embeddedness in the world. As cultural analysts and software studies scholars, we posit that while an executing practice may be composed and propelled by the programmed drives and demands it instantiates, it must, nonetheless, necessarily negotiate the uncertainties of its encounters with other live processes, drives, and materials. These uncertain encounters of execution might include the particular nature of data and their varying levels of machine readability, the time-sensitive negotiations of computer networks, or the various materialities and affectivities that underpin and afford various forms of computational practices. They will also involve formative human-influenced factors—factors that can be of both a predictable and an unpredictable nature. Long-standing hegemonies of various kinds (e.g., capitalism, white and male power structures) become ingrained within the present manifestations of computational culture, manifested in the boardrooms and engineering teams of many dominant computational platforms of the present (Fowler 2017; Matsakis 2017; Miley 2015).

This constant negotiation of both seemingly certain and uncertain elements that arise points to the vitality and also urgency of this quality of the ongoing, lively encounters of forms of execution in the world. In an admiring moment of reflection in his “Computing Machinery and Intelligence” essay, mathematician and computing pioneer Alan Turing (1950, 450) comments on the way in which “machines take me by surprise with great frequency.” While admitting that this partly reflects his own tendency to work quickly and risk the inevitable calculative inaccuracies and missteps that come with such a practical desire to get on with things, Turing is highlighting the need to take seriously the generative quality of experiment and the impossibility of being able to fully predict in advance both the machinations and possibilities of computation and its active executions. At the same time, as one sees in Turing’s own fate (Halberstam 1998) or the resurgence of antifeminist, antiblack, anti-immigrant, and other populist backlashes in contemporary cultures, machines can readily afford, feed back, and amplify certain dispositions and energies of their creators, users, and specific contexts in which they come into being and take hold. This is despite a radical openness to reconfiguration in the very nature of their makeup, an openness to alterity that continues to undermine forms of certainty with what is in computing a foundational acknowledgment of the uncertain.

To Halt or Not to Halt

Why might the notion of execution be understood as central to the concepts of not only computing but also computational culture more widely? What are the uncertainties arising from the execution of instructions? It is common knowledge that a computer program executes, meaning that the physical machine performs certain actions according to the logic of a given program. It is also common knowledge to suppose that this program had to be previously written in symbolic human-readable code (source code), which was compiled into machine-readable instructions and formatted (for instance, an “.exe” file extension) to produce the program as such. Thus, the notion of execution speaks to this intricate conjugation of code, instructions, and actions, alluding to the process by which a symbolic order (data-as-program) instructs a machine how to act or perform.

In theoretical computer science, it is conventional to talk in abstract terms of a finite sequence of instructions to denote an algorithm or a program. It is assumed that this sequence will be “fed” to a machine in order to execute and, in turn, yield expected results or a valid “output.” The moment of execution as such is thus tacitly encapsulated within a well-defined instructions-result coupling (cause and effect), in fact producing an instructions-(execution)-result triad rather than a mere oversimplified couple. We know from Turing and his contemporary mathematician Alonzo Church that there exist sequences of instructions that will never produce any output whatsoever, causing the machine to enter an infinite loop, so to speak—executing ad vitam aeternam. In other words, while the sequence of instructions fed to the machine may well be finite, its machine execution can nonetheless be infinite. No mathematical guarantees or proofs can state that a given, finite set of instructions will inevitably produce a finite number of execution steps. This problem is, mathematically speaking, undecidable.

As a problematic or limit, this type of “in-finite” is at the core of the notion of computation and is usually referred to as the *halting problem*. In fact, one of the main definitions of computation or computability is that it is a task that needs to terminate or a process that has to stop (Kleene 1952, chapter 13, §67). What is dubbed “compute-able” is a process that terminates with an output, while a process that does not is judged “un-compute-able.” We can clearly see how compute-ability, as a definition, is intermingled with the notion of time, here taking the form of a delay or a waiting period. Given that one can define an executable physical process as a procedure “that a finite observer can set in motion to generate the values of a desired function until it generates a readable result” (Piccinini 2011, 741; italics added), one may be inclined to ask: Wait until when?

This formative issue of uncertainty in any executable procedure points to a profound and abstruse stasis at the heart of computation—that is, within the aforementioned instructions-(execution)-result conjugate. When instructions are fed to a machine in order to be executed, there is theoretically no guarantee the machine will produce any viable results in due time. This odd state of waiting, as a suspense or an anxious differing of time, is at the very core of mathematical theories and practices of computing—and must have been felt badly in the early days of computing, when slow machines carried out execution (hence the need to devise cunning debugging tactics). While our machines may have become faster over the years, this fretful computational wait is nonetheless still effective, and that by definition. When data are understood as program, execution speaks not only to the uncertain but also to the undecidable.

Query {the Logic of Request && Response};

Yet data can, obviously, be understood as data, be they big or small. Data as input or output of a given program or platform can be framed by the notion of query. In today’s big-data regimes, extracting data from online or social media platforms becomes one of the most important methods in computational culture and a practice in which the concept of query is critical (Soon 2016, 132; Snodgrass & Soon 2019). The concept and method of querying databases are concerned with selecting and extracting specific data records with both inputs and outputs. The execution of a query is a two-way communication process that is both a request and a response. A query has the capacity to specify, create, and identify relations through the logic of request and response, symbolized by the SQL “SELECT” syntax used in database systems. A networked program necessarily needs to wait for the response to its queried request. This waiting is precisely uncertain not just as a mathematical problem but also as a practical execution in a big-data regime where a database is subjected to the complexity of the query itself, the configurations of machines, the geographic locations of servers, the number of records addressed by the query, and many other infrastructural parameters beyond the undecidable logic of computation.

Web queries have become a standard package of social media platforms, at least in the case of the major companies across the Eastern and Western continents, including but not limited to Google, Facebook, Amazon, Sina Weibo, WeChat, and Twitter. One of the important aspects of dynamic web services, according to Tim O’Reilly (2005), who popularized the term Web 2.0, is how data querying and management allow a form of “remixability”—that is, to “remix the data into new services.” This type of data remixability includes not only the capturing, storing, and organizing of data but also their spatial redistribution. As a result, many websites have shifted their paradigm from the concept of single “sites” to programmable “platforms” (Helmond 2015, 35). People can now program, build, and extend the offering of the platform and redistribute services via the availability of web queries, also known as application programming interfaces (APIs). These programming interfaces enable the execution of queries on various distinct sites (spatial dimensions) and at various asynchronous times (temporal dimensions).

Today, many APIs are free to use but feature certain practical and epistemological restrictions. Most platforms focus on technical and prescriptive usage of their APIs but do not expose how data are actually being queried within their private guarded databases. Without a visible “algorithmic” picture, it is extremely difficult to grasp the priorities and logics involved in modern data queries. Important questions arise from such an opaque relation: What are the assumptions behind highly structured APIs? What is prioritized and, consequently, what is included and excluded from the responded data format? Queries exhibit a certain material power that executes the inclusion and exclusion of specific types and ranges of data. When we request search results from Google via its API, for example, what do Google’s algorithms mean by the “most relevant” or top ten search results? What are the relationships between the search results, the business decisions, and the individuals’ locations, histories, and preferences?[[1]](#footnote-1)

The answers to these questions, particularly in the present moment of opaque platform politics, are largely hidden. Black-boxed according to a variety of computational, economic, and political reasons, such types of information hiding spread their particular forms of illegibility, partial knowledges, or outright unknowns in a way that infuses the network with a pervasive sense of ongoing uncertainty. While parameters and decisions can be changed and updated in a seamless way, the state and logic of platforms are never transparent and static. The aforementioned sense of anxious waiting, which lies at the heart of the definition of computation and its executions, is in such instances overlaid with this uncertainty about the composition of queries for big-data regimes—queries that we have become productively accustomed to yet highly dependent upon. In this way, executing a query is an uncertain act that conflates languages, symbols, meanings, geopolitical decisions, and dynamic, uncertain consequences.

#metoo

Not all uncertainty comes from the machine or is generated (yet) by the calculations performed by machines and their captured data. Uncertainty also inhabits affective data bodies (Tyżlik-Carver 2017) that reveal big-data’s inability to function without them. An affective data body is the other that has been left out, a material witness (Schuppli 2018) that does not return computational results, showing that not all count the same, and some do not count at all unless they are left to be exploited; not beautiful and structured data but messy, dirty, and hard to represent, let alone compute. If queried data concern the material powers of inclusion and exclusion, affective data are then registered by the body left in pain, or left numb, or overworked, or tired, or unwanted, or dead. Or it is a body healed, or both in pain and healed. Affective data manifest in each body that struggles and waits to “become eligible for recognition” (Butler 2009, iv).

Still, affective data can be captured and shoved into the wide-open mouth of big-data regimes. They can be simultaneously tagged by each subject using the hashtag syntax to communicate and register that #icantbreathe and #metoo. It is at the point of the critical-mass flooding of social media communication channels that affective data are returned as big-data queries and metadata callout: the other of surveillance, reclaiming data tagging to register abuse. The #metoo campaign can be thought of as a query for empowerment and empathy. Originally initiated by Tarana Burke as the me tooMovement™ ten years before the 2017 campaign and before the # hashtag was even a thing, the campaign was meant to build a coalition and to give young women the power of empathy and recognition that they are not alone in their experience of sexual assault, abuse, and exploitation (Justbeinc 2017; Vagianos 2017; West 2017).

Alyssa Milano’s tweeting #metoo optimized this decade-old campaign, returning 1.7 million tweets within a week and reaching twelve million Facebook posts in twenty-four hours (CBS/Associated Press 2017; Park 2017). Is this optimization a glitch in the system that otherwise has sustained racist and misogynist normcore off-line and online while keeping others silenced? Or does it again illustrate how some voices are heard above others? Accountable to the original cause, however, the #metoo campaign is a contemporary intervention into online and off-line spaces of (white) male hegemony in power structures. It is both a computational and affective query that reveals affinities between women of color, nonbinary people, white women, and computational infrastructures whose relations seem only contingent upon the uncertainties of affective data bodies as they intervene in systems of data power by saying #metoo and waiting to be heard through the algorithmic display of social media feeds. Uncertainties of the computational and human kind lead to formations of a variety of constellations and cultures that result from tragic events or cases of abuse, from Gamergate to #blacklivesmatter, #metoo, and others to come.

Exit(0);

As we have consistently argued here and elsewhere (Pritchard, Snodgrass, and Tyżlik-Carver 2017), execution can be thought of as a continual resolution, a composing and rendering of various generative, undecidable, invisible, affective, and intersecting uncertainties that unfold within the context of present-day computational cultures and infrastructures. The seamlessly ungraspable relations that result from executions are concealed within a conflation of languages, symbols, and meanings with technical, cultural, and political decisions. As the term *execution* may suggest, this conflation constitutes a form of violence at once effected and effaced, disguised under a moniker of lawful, interactive rationality.

Even though computational systems seem to perform in a purely deterministic, axiomatic, and strict way, or at least seem to be trying to, they actually do not. Contrary to the common-sense notion that “computational systems are formal axiomatic systems” (Fazi 2018, 98), which would imply that any computational claims can be directly deduced from a set of immutable axiomatic “truths,” we rather think of computation as fundamentally uncertain and undecidable. Assuredly, computation can be said to be formal (in the sense that a programming language is), yet it is not axiomatic. As such, being formal, prescriptive, and even tyrannical (Gauthier 2017) does not preclude computation from being uncertain. As in the cases of the halting problem, big-data query, and the #metoo campaign, we see that formal systems and affective bodies already expose their own indeterminacy, necessarily. As stated above, there is no certainty that a finite set of instructions given to a machine (be it SQL statements, Siri voice commands, x86 instructions, or otherwise) will inevitably produce a finite result in due time. Beyond possible infinite loops, the definition of computability is itself commingled with the notion of time, and it is for this reason that contemporary computer scientists and mathematicians talk about computational heuristics rather than computational axiomatics. It is through this notion of perpetual wait that the concept of execution speaks to the uncertain: waitingforGodot.exe.

References

Butler, Judith. 2009. “Performativity, precarity and sexual politics.” AIBR, Revista de Antropolgia Iberoamericana 4 (3): i–xiii.

CBS/Associated Press. 2017. “More than 12m ‘me too’ Facebook posts, comments, reactions in 24 hours.” CBS News, October 17, 2017. https://www.cbsnews.com/news/metoo-more-than-12-million-facebook-posts-comments-reactions-24-hours/.

Fazi, Beatrice. 2018. *Contingent Computation: Abstraction, Experience, and Indeterminacy in Computational Aesthetics*. Lanham: Rowman & Littlefield International.

Fowler, Susan J. 2017. “Reflecting on one very, very strange year at Uber.” Susan Fowler (blog). February 19, 2017. https://www.susanjfowler.com/blog/2017/2/19/reflecting-on-one-very-strange-year-at-uber.

Gauthier, David. 2018. “On commands and executions: Tyrants, spectres and vagabonds.” In Executing Practices, edited by Helen Pritchard, Eric Snodgrass, and Magdalena Tyżlik-Carver, 69–84. Data Browser 06/Open Humanities Press.

Halberstam, Judith. 1998. “Automating gender: Postmodern feminism in the age of the intelligent machine.” In Sex/ Machine: Readings in Culture, Gender, and Technology, edited by Patrick D. Hopkins, 468–483. Bloomington: Indiana University Press.

Helmond, Anne. 2015. “The web as platform: Data flows in social media.” PhD diss., University of Amsterdam. http://www.annehelmond.nl/wordpress/wpcontent/uploads//2015/08/Helmond\_WebAsPlatform.pdf.

Justbeinc. 2017. “The movement.” http://justbeinc.wixsite.com/justbeinc/the-me-too-movement.

Kleene, Stephen Cole. 1952. Introduction to Metamathematics. Amsterdam: North-Holland.

Matsakis, Louise. 2017. “Google employee’s anti-diversity manifesto goes ‘internally viral.’” Motherboard, August 5, 2017. https://motherboard.vice.com/en\_us/article/kzbm4a/employees-anti-diversity-manifesto-goes-internally-viral-at-google.

Miley, Leslie. 2015. “Thoughts on diversity part 2: Why diversity is difficult.” Tech Diversity Files, November 3, 2015. https://medium.com/tech-diversity-files/thought-on-diversity-part-2-why-diversity-is-difficult-3dfd552fa1f7.

O’Reilly, Tim (website). 2005. “What is Web 2.0.” September 30, 2005. http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html.

Park, Andrea. 2017. “#metoo reaches 85 countries with 1.7 million tweets.” CBS News, October 24, 2017. https://www.cbsnews.com/news/metoo-reaches-85-countries-with-1-7-million-tweets/.

Piccinini, Gualtiero. 2011. “The physical Church-Turing thesis: Modest or bold?” British Journal for the Philosophy of Science 62 (4): 733–769.

Pritchard, Helen, Eric Snodgrass, and Magdalena Tyżlik-Carver, eds. 2018. Executing Practices. Data Browser 06/Open Humanities Press.

Samson, Audrey, and Winnie Soon. 2015. “Network Affordances: Unpredictable parameters of a Hong Kong SPEED SHOW.” *The Fibreculture Journal* 24. http://twentyfour.fibreculturejournal.org/2015/06/04/44/.

Schuppli, Susan. 2018. Material Witness: Forensic Media and the Production of Evidence. Cambridge, MA: MIT Press.

Soon, Winnie. 2016. “Executing liveness: An examination of the live dimension of code inter-actions in software (art) practice.” PhD diss., University of Aarhus, Denmark.

Snodgrass, Eric & Soon, Winnie. 2019. “API practices and paradigms: Exploring the protocological parameters of APIs as key facilitators of sociotechnical forms of exchange” *First Monday* 24 (2).

Turing, Alan M. 1950. “Computing machinery and intelligence.” Mind 59 (236): 433–460.

Tyżlik-Carver, Magdalena. 2018. “Posthuman curating and its biopolitical executions: The case of curating content.” In Executing Practices, edited by Helen Pritchard, Eric Snodgrass, and Magdalena Tyżlik-Carver, 171–190. Data Browser 06/Open Humanities Press.

Vagianos, Alanna. 2017. “The ‘me too’ campaign was created by a black woman 10 years ago.” Huffington Post, October 17, 2017. https://www.huffingtonpost.com/entry/the-me-too-campaign-was-created-by-a-black-woman-10-years-ago\_us\_59e61a7fe4b02a215b336fee.

West, Lindy. 2017. “Yes, this is a witch hunt. I’m a witch and I’m hunting you.” New York Times, October 17, 2017. <https://www.nytimes.com/2017/10/17/opinion/columnists/weinstein-harassment-witchunt.html>.

1. See a full discussion on the entanglement of affordances in: A. Samson, and W. Soon. “Network Affordances: Unpredictable parameters of a Hong Kong SPEED SHOW.” *The Fibreculture Journal* 24 (2015). http://twentyfour.fibreculturejournal.org/2015/06/04/44/. [↑](#footnote-ref-1)